

```
*****
;
; PROGRAM ID: FORMAT
;
*****
;
; PRESENTED BY: JADE COMPUTER PRODUCTS
; 4901 W. ROSECRANS BLVD.
; HAWTHORNE, CALIFORNIA
; 90250, U.S.A.
;
*****
;
; VERSION: CP/M 2.2 RELEASE 2A
;
*****
;
; WRITTEN BY: STAN KRUMME
;
*****
; FORMAT IS A SYSTEM UTILITY WHICH PROVIDES A MEANS
; TO WRITE A SINGLE OR DOUBLE DENSITY FORMAT ON ANY
; OF DRIVES A THROUGH D. THIS UTILITY ALSO PROVIDES
; A COPY-SYSTEM-TRACKS FEATURE. THIS IS A USEFUL
; FUNCTION FOR FORMAT AS THE SYSTEM TRACKS CAN BE
; WRITTEN WITH THE OPERATING SYSTEM WHEN FORMATTED.
; FORMAT IS 8080/8085/Z80 COMPATABLE.
;
*****
;
; FORMAT INJECTION MODULES ARE COMMAND COMPATABLE WITH*
; THE FOLLOWING WESTERN DIGITAL CONTROLLER CHIPS. *
; DOUBLE D USER SWITCH 0 (U0 OR R0) MUST BE SET TO *
; INDICATE THE CONTROLLER CHIP DATA BUS POLARITY. *
;
; CONTROLLER IC USER SWO
; -----
; FD1791-02 (01) CLOSED
; FD1793-02 (01) OPENED
; FD1795-02 CLOSED
; FD1797-02 OPENED
;
*****
;
; RELEASE 2A: SINGLE AND DOUBLE SIDED DRIVES CAN BE *
; FORMATED. INSPECTION OF TWO SIDED* SIGNAL FROM THE *
; DISK DRIVE DETERMINES NUMBER OF SIDES. WITH DOUBLE *
; SIDED DISKETTES, BOTH SIDES FORM ONE LOGICAL DISK. *
; EACH DOUBLE DENSITY TRACK NOW CONTAINS 50 SECTORS. *
;
*****
```

```

;*****
; DRIVER MODULE DEFINITIONS
;*****

```

```

000A      LF      ==      00AH      ;ASCII LINE FEED.
000D      CR      ==      00DH      ;CARRIAGE RETURN.
0024      EOM      ==      '$'      ;STRING TERMINATOR.
0100      TPA      ==      0100H     ;TRANSIENT PROGRAM.
0000      TRK.0    ==      0         ;TRACK 0.
0001      TRK.1    ==      1         ;TRACK 1.
0002      TRK.2    ==      2         ;TRACK 2.
0080      SEC.SZ   ==      128       ;128 BYTES PER SECTOR.
0001      ID.SEC   ==      1         ;ID SECTOR NUMBER.
0000      REBOOT   ==      0         ;REBOOT ADDRESS.
0001      BS.PTR   ==      0001H     ;WARM ADDR POINTER.
0001      NO.LOG   ==      01H       ;REQUEST NO LOG-ON.
00FE      FT.ERC   ==      1111110B ;FORMAT ERROR MASK.
0001      FT.TSM   ==      00000001B ;TWO SIDED MASK.

```

```

;*****
; INJECTION MODULE DEFINITIONS
;*****

```

```

1700      FMT.EA   ==      1700H     ;FORMAT EXEC ADDRESS.
0005      WD.TRK   ==      005H      ;DOUBLE D TRACK PORT.
0007      WD.DTA   ==      007H      ;DOUBLE D DATA PORT.
0080      XP.DSH   ==      80H       ;DATA SYNC HOLD PORT.
0000      ZEROS    ==      00000000B ;ALL ZERO BYTE.
00FF      ONES     ==      11111111B ;ALL ONES BYTE.

```

```

;*****
; BDOS CALL - VECTOR NUMBERS
;*****

```

```

0005      BDOS     ==      0005H     ;SYSTEM CALL ADDR.
0009      BC.PTX   ==      009H      ;PRINT STRING CONSOLE.
000A      BC.RCB   ==      00AH      ;READ CONSOLE BUFFERD.

```

```

;*****
; ASSEMBLER DIRECTIVES
;*****

```

```

                                .I8080
                                .FABS
                                .PHEX
                                .XLINK
0100      .LOC      TPA

```

```

;*****

```



```

; *****
; PROGRAM BEGINS
; *****

0100      C3 0146      BEGIN:  JMP      INIT          ;GO TO INITIALIZE.

; *****
; ASCII IDENTIFICATION INSERT
; *****

0103      4A4144452043      .ASCII  'JADE COMPUTER PRODUCTS '
011A      444F55424C45      .ASCII  'DOUBLE D - FORMAT 8" '
012F      56455253494F      .ASCII  'VERSION 2.2 RELEASE 2A '

; *****
; SET STACK POINTER AND ISSUE LOG-ON
; *****

0146      31 0500      INIT:   LXI      SP,SP.TOP      ;SET STACK POINTER.
0149      11 0716      LXI      D,MSG.BG      ;LOAD MESSAGE ADDR.
014C      CD 0297      CALL     MSG.OT      ;ISSUE MESSAGE.

; *****
; LOAD BIOS VECTORS JUMP TABLE - WARM THRU FORMAT
; *****

014F      01 0033      LXI      B,BS.VSZ      ;SET BIOS VECTORS SIZE.
0152      11 02A4      LXI      D,BS.WRM      ;SET FORMAT TABLE.
0155      2A 0001      LHLD     BS.PTR      ;WARM VECTOR POINTER.
0158      CD 02D7      CALL     B.MOVE      ;BLOCK MOVE VECTORS.

; *****
; SELECT DRIVE TO FORMAT ON
; *****

015B      21 0781      LXI      H,MSG.FD      ;FORMAT ON DRIVE MSG.
015E      CD 0485      CALL     SEL.DR      ;CALL SELECT DRIVE.
0161      32 04DF      STA      FD.NBR      ;FORMAT DRIVE NMBR.

; *****

```

```

; *****
; DISPLAY FUNCTIONS LIST
; *****

```

```

0164      11 07BB      LIST:   LXI      D,MSG.FL      ;FUNCTIONS MSG ADDR.
0167      CD 0297      CALL    MSG.OT      ;ISSUE THIS MESSAGE.

```

```

; *****
; INQUIRE SELECTION
; *****

```

```

016A      11 08F4      SELECT: LXI      D,MSG.SF      ;SELECT FUNCTION MSG.
016D      CD 0297      CALL    MSG.OT      ;ISSUE THIS MESSAGE.
0170      CD 029C      CALL    CNS.IN      ;GET CONSOLE CHARACTER.
0173      3A 0501      LDA     RC.NBR      ;LOAD BUFFER SIZE.
0176      FE01          CPI     1          ;CHECK FOR 1 CHARACTER.
0178      C2 016A      JNZ     SELECT      ;OTHER THAN 1 TOO BAD.

```

```

; *****
; SELECT FUNCTION DRIVER
; *****

```

```

017B      3A 0502      LDA     RC.TXT+0      ;LOAD CONSOLE CHAR.
017E      FE31          CPI     "1"
0180      CA 01AA      JZ      FUN.1          ;FMT DOUBLE DENSITY.
0183      FE32          CPI     "2"
0185      CA 0208      JZ      FUN.2          ;FMT SINGLE DENSITY.
0188      FE33          CPI     "3"
018A      CA 01D2      JZ      FUN.3          ;FMT 3740.
018D      FE34          CPI     "4"
018F      CA 01EB      JZ      FUN.4          ;READ SYSTEM TRACKS.
0192      FE35          CPI     "5"
0194      CA 025D      JZ      FUN.5          ;WRITE SYSTEM TRACKS.
0197      FE2A          CPI     "*"
0199      CA 023B      JZ      FMT.ST        ;FORMAT SYSTEM TRACKS.
019C      FE27          CPI     "/"
019E      CA 04CC      JZ      RST.7          ;DDT TRAP.

```

```

; *****
; MUST BE A BAD CHOICE
; *****

```

```

01A1      11 07A2      LXI     D,MSG.SE      ;SELECT ERROR MESSAGE.
01A4      CD 0297      CALL    MSG.OT      ;ISSUE MESSAGE.
01A7      C3 016A      JMP     LIST        ;DIPLAY LIST AGAIN.

```

```

; *****

```

```
*****
; FUNCTION 1 - FORMAT IN DOUBLE DENSITY
*****
```

```
01AA 3A 0647 FUN.1: LDA DD.FLG ;LOAD DDENS FLAGS.
01AD 32 04DC STA F.FLAG ;STORE FORMAT FLAGS.
01B0 3E00 MVI A,TRK.0 ;TRACK 0.
01B2 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
01B5 CD 031B CALL FMT.SD ;FORMAT TRACK SDENS.
01B8 C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
01BB 3E01 MVI A,TRK.1 ;TRACK 1 VALUE.
01BD 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
01C0 CD 0321 ..REPT: CALL FMT.DD ;FORMAT TRACK DDENS.
01C3 C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
01C6 CD 0279 CALL TRK.NX ;SET FOR NEXT TRACK.
01C9 CA 01C0 JZ ..REPT ;FORMAT NEXT TRACK.
01CC CD 02F2 ..ID: CALL WDD.ID ;WRITE DDENS ID SECTOR.
01CF C3 016A JMP SELECT ;SELECT NEW FUNCTION.
```

```
*****
; FUNCTION 3 - FORMAT STANDARD 3740 - SINGLE SIDED
*****
```

```
01D2 3E00 FUN.3: MVI A,0 ;3740 SDENS FLAGS.
01D4 32 04DC STA F.FLAG ;STORE FORMAT FLAGS.
01D7 3E00 MVI A,TRK.0 ;TRACK 0.
01D9 32 04D9 STA TRK.NO ;SET TRACK NUMBER.
01DC CD 031B ..REPT: CALL FMT.SD ;FORMAT TRACK SDENS.
01DF C2 028E JNZ TRK.ER ;JUMP ERROR DETECTED.
01E2 CD 0279 CALL TRK.NX ;SET FOR NEXT TRACK.
01E5 CA 01DC JZ ..REPT ;FORMAT NEXT TRACK.
01E8 C3 016A JMP SELECT ;SELECT NEW FUNCTION.
```

```
*****
; FUNCTION 4 - READ SYSTEM TRACKS
*****
```

```
01EB 21 090E FUN.4: LXI H,MSG.RS ;READ DRIVE MSG.
01EE CD 0485 CALL SEL.DR ;SELECT READ SYS DRV.
01F1 3E52 MVI A,'R' ;READ TRANSFER CODE.
01F3 32 04D4 STA TF.DIR ;SET TRANSFER DIRC.
01F6 32 04DD STA SYS.RF ;SET SYSTEM READ FLAG.
01F9 CD 0355 CALL TRNSFR ;READ SYSTEM TRACKS.
01FC 3A 04DF LDA FD.NBR ;GET FORMAT DRV NMBR.
01FF 4F MOV C,A ;PUT INTO C REG.
0200 1E01 MVI E,NO.LOG ;INSURE NO LOGON.
0202 CD 02BC CALL BS.DSK ;BIOS SELECT DISK.
0205 C3 016A JMP SELECT ;RESELECT FUNCTION.
```

```
*****
```

FORMAT - JADE DOUBLE D

FUNCTION CONTROLLERS

```

;*****
; FUNCTION 2 - FORMAT IN SINGLE DENSITY *
;*****

```

```

0208      3A 0547      FUN.2:  LDA      SD.FLG      ;LOAD SDENS FLAGS.
020B      32 04DC      STA      F.FLAG      ;STORE FORMAT FLAGS.
020E      3E00          MVI      A,TRK.0      ;TRACK 0.
0210      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
0213      CD 031B      CALL     FMT.SD      ;FORMAT TRACK SDENS.
0216      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
0219      3E01          MVI      A,TRK.1      ;TRACK 1
021B      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
021E      CD 0321      CALL     FMT.DD      ;FORMAT TRACK DDENS.
0221      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
0224      3E02          MVI      A,TRK.2      ;TRACK 2.
0226      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
0229      CD 031B      ..REPT: CALL     FMT.SD      ;FORMAT TRACK SDENS.
022C      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
022F      CD 0279      CALL     TRK.NX      ;SET FOR NEXT TRACK.
0232      CA 0229      JZ       ..REPT      ;FORMAT NEXT TRACK.
0235      CD 02E2      ..ID:  CALL     WSD.ID      ;WRITE SDENS ID SECTOR.
0238      C3 016A      JMP      SELECT      ;SELECT NEW FUNCTION.

```

```

;*****
; NON DOCUMENTED FUNCTION - FORMAT JADE SYSTEM TRACKS *
;*****
; USED FOR SPECIAL PURPOSE - NOT NEEDED BY END USER *
;*****

```

```

023B      3A 0547      FMT.ST: LDA      SD.FLG      ;LOAD SDENS FLAGS.
023E      32 04DC      STA      F.FLAG      ;STORE FORMAT FLAGS.
0241      3E00          MVI      A,TRK.0      ;TRACK 0.
0243      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
0246      CD 031B      CALL     FMT.SD      ;FORMAT TRACK SDENS.
0249      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
024C      3E01          MVI      A,TRK.1      ;TRACK 1
024E      32 04D9      STA      TRK.NO      ;SET TRACK NUMBER.
0251      CD 0321      CALL     FMT.DD      ;FORMAT TRACK DDENS.
0254      C2 028E      JNZ      TRK.ER      ;JUMP ERROR DETECTED.
0257      CD 02E2      CALL     WSD.ID      ;WRITE SDENS ID SECTOR.
025A      C3 016A      JMP      SELECT      ;SELECT NEW FUNCTION.

```

```

;*****

```

```
*****
; FUNCTION 5 - WRITE SYSTEM TRACKS
*****
```

```
025D 3A 04DD FUN.5: LDA SYS.RF ;LOAD SYSTEM READ FLAG.
0260 FE52 CPI 'R' ;TEST IF READ CODE.
0262 C2 0270 JNZ ..NSYS ;JUMP IF NO SYSTEM.
0265 3E57 MVI A,'W' ;WRITE TRANSFER CODE.
0267 32 04D4 STA TF.DIR ;SET TRANSFER DIRC.
026A CD 0355 CALL TRNSFR ;WRITE SYSTEM TRACKS.
026D C3 016A JMP SELECT ;WRITE ANOTHER DISK.
0270 11 095E ..NSYS: LXI D,MSG.NR ;NO SYSTEM LOADED MSG.
0273 CD 0297 CALL MSG.OT ;ISSUE THIS MESSAGE.
0276 C3 016A JMP SELECT ;SELECT NEW FUNCTION.
```

```
*****
; NEXT TRACK SELECT ROUTINE
*****
```

```
0279 3A 04DA TRK.NX: LDA TRK.MX ;LOAD MAX TRACK NMBR.
027C 47 MOV B,A ;SAVE IN REG B.
027D 3A 04D9 LDA TRK.NO ;GET THIS TRACK NO.
0280 B8 CMP B ;CHECK FOR LAST TRACK.
0281 CA 028A JZ ..DONE ;JUMP IF LAST TRACK.
0284 3C INR A ;GET NEXT TRACK.
0285 32 04D9 STA TRK.NO ;STORE NEXT TRACK.
0288 AF XRA A ;SET ZERO FLAG.
0289 C9 RET ;RETURN TO CALLER.
028A 3EFF ..DONE: MVI A,ONES ;SET ALL ONES.
028C A7 ANA A ;SET FLAG NOT ZERO.
028D C9 RET ;LAST TRACK EXIT.
```

```
*****
; FORMAT TRACK ERROR
*****
```

```
028E 11 0945 TRK.ER: LXI D,MSG.FE ;FORMAT ERROR MSG ADDR.
0291 CD 0297 CALL MSG.OT ;DISPLAY MESSAGE.
0294 C3 016A JMP SELECT ;SELECT NEW FUNCTION.
```

```
*****
```

```

;*****
; MESSAGE DISPLAY ROUTINE
;*****

0297 0E09 MSG.OT: MVI C,BC.PTX ;PRINT TEXT VECTOR.
0299 C3 0005 JMP BDOS ;CONTINUE IN BDOS.

;*****
; CONSOLE INPUT ROUTINE
;*****

029C 11 0500 CNS.IN: LXI D,RC.BUF ;KEYBOARD BUFFER ADDR.
029F 0E0A MVI C,BC.RCB ;BDOS CONSOLE BUF READ.
02A1 C3 0005 JMP BDOS ;CONTINUE IN BDOS.

;*****
; BIOS VECTOR DEFINITIONS
;*****

02A4 C3 0000 BS.WRM: JMP 0 ;RELOAD CCP/BDOS.
02A7 C3 0000 JMP 0 ;GET CONSOLE STATUS.
02AA C3 0000 JMP 0 ;CONSOLE CHAR INPUT.
02AD C3 0000 JMP 0 ;CONSOLE CHAR OUTPUT.
02B0 C3 0000 JMP 0 ;PRINTER OUTPUT.
02B3 C3 0000 JMP 0 ;PUNCH CHARACTER OUT.
02B6 C3 0000 JMP 0 ;READER INPUT.
02B9 C3 0000 JMP 0 ;HOME SELECTED DRIVE.
02BC C3 0000 BS.DSK: JMP 0 ;SELECT DISK DRIVE.
02BF C3 0000 BS.TRK: JMP 0 ;SET TRACK NUMBER.
02C2 C3 0000 BS.SEC: JMP 0 ;SET SECTOR NUMBER.
02C5 C3 0000 BS.DMA: JMP 0 ;SET TRANSFER ADDR.
02C8 C3 0000 BS.RDS: JMP 0 ;READ DISK SECTOR.
02CB C3 0000 BS.WRS: JMP 0 ;WRITE DISK SECTOR.
02CE C3 0000 JMP 0 ;LIST DEV STATUS.
02D1 C3 0000 JMP 0 ;SECTOR TRANSLATE.
02D4 C3 0000 BS.FMT: JMP 0 ;FORMAT DISK TRACK.
0033 BS.VSZ == .-BS.WRM ;CALCULATE SIZE.

;*****
; BLOCK MOVE SUBROUTINE
;*****

02D7 7E B.MOVE: MOV A,M ;GET BYTE
02D8 23 INX H ;INC SOURCE.
02D9 12 STAX D ;STORE BYTE.
02DA 13 INX D ;INC DESTINATION.
02DB 0B DCX B ;ONE LESS TO DO.
02DC 78 MOV A,B ;GET B REG.
02DD B1 ORA C ;OR IN C REG.
02DE C2 02D7 JNZ B.MOVE ;REPEAT FOR LENGTH.
02E1 C9 RET ;RETURN CALLER.

;*****

```



```
*****
; WRITE ID SECTOR
*****
```

```
***** ( SET TRANSFER ADDRESS ) *****
```

02E2	01 0516	WSD.ID:	LXI	B,IDS.SS	;ID SECTOR ADDRESS.
02E5	3A 04D8		LDA	TS.FLG	;GET TWO SIDES FLG.
02E8	A7		ANA	A	;TEST.
02E9	CA 02FF		JZ	WRT.ID	;JUMP ONE SIDED.
02EC	01 0596		LXI	B,IDS.DS	;DOUBLE SIDED.
02EF	C3 02FF		JMP	WRT.ID	;JUMP TWO SIDED.
02F2	01 0616	WDD.ID:	LXI	B,IDS.SD	;ID SECTOR ADDRESS.
02F5	3A 04D8		LDA	TS.FLG	;GET TWO SIDES FLG.
02F8	A7		ANA	A	;TEST.
02F9	CA 02FF		JZ	WRT.ID	;JUMP ONE SIDED.
02FC	01 0696		LXI	B,IDS.DD	;DOUBLE SIDED.
02FF	CD 02C5	WRT.ID:	CALL	BS.DMA	;BIOS TRANSFER ADDR.

```
***** ( SET TRACK AND SECTOR NUMBERS ) *****
```

0302	0E00		MVI	C,TRK.0	;TRACK 0 SET.
0304	CD 02BF		CALL	BS.TRK	;BIOS SET TRACK.
0307	0E01		MVI	C,ID.SEC	;ID SECTOR VALUE.
0309	CD 02C2		CALL	BS.SEC	;BIOS SET SECTOR.

```
***** ( PERFORM WRITE SECTOR ) *****
```

030C	CD 02CB		CALL	BS.WRS	;BIOS WRITE SECTOR.
030F	B7		ORA	A	;SET CONDITION CODES.
0310	C8		RZ		;RETURN USER GOOD.
0311	11 092B		LXI	D,MSG.NC	;TRANSFER INCOMPLETE.
0314	CD 0297		CALL	MSG.OT	;ISSUE MESSAGE.
0317	3EFF		MVI	A,ONES	;SET ACUMULATOR.
0319	A7		ANA	A	;SET FLAGS NOT ZERO.
031A	C9		RET		;ERROR RETURN.

```
*****
```



```

;*****
;  FORMAT TRACK DRIVER
;*****

;***** ( DENSITY ENTRIES )*****

031B      01 0A00      FMT.SD: LXI      B,FT3740      ;LOAD INJECTION ADDR.
031E      C3 0324      JMP      ST.DMA      ;GO SET DMA ADDR.
0321      01 0B00      FMT.DD: LXI      B,FTJ50D      ;LOAD INJECTION ADDR.

;***** ( SET INJECTION MODULE ADDRESS )*****

0324      CD 02C5      ST.DMA: CALL     BS.DMA      ;SET TRANSFER ADDRESS.

;***** ( SET TRACK NUMBER AND DCM FLAGS )*****

0327      3A 04D9      LDA      TRK.NO      ;LOAD TRACK NMBR.
032A      4F          MOV      C,A          ;PUT INTO C REGISTER.
032B      CD 02BF      CALL     BS.TRK      ;SET TRACK NMBR.

032E      3A 04DC      LDA      F.FLAG      ;LOAD DCM FLAG.
0331      4F          MOV      C,A          ;DCM FLAGS.
0332      CD 02C2      CALL     BS.SEC      ;SET DCM FLAGS.

;***** ( PERFORM FORMAT TRACK )*****

0335      CD 02D4      CALL     BS.FMT      ;BIOS WRITE TRACK.
0338      32 04D7      STA      FT.STS      ;FORMAT STATUS.
033B      E6FE      ANI      FT.ERC      ;TEST FOR ERRORS.
033D      C0          RNZ          ;ERROR EXIT.

;***** ( SET CONTROLS FOR SIDE/SIDES )*****

033E      3A 04D7      LDA      FT.STS      ;GET STATUS.
0341      E601      ANI      FT.TSM      ;TEST TWO SIDES FLAG.
0343      32 04D8      STA      TS.FLG      ;STORE FLAG.
0346      C2 034E      JNZ      ..TWSD      ;TWO SIDES IS A 1.
0349      3E4C      MVI      A,77-1      ;SINGLE SIDED MAX.
034B      C3 0350      JMP      ..EXIT      ;EXIT.
034E      3E99      ..TWSD: MVI      A,2*77-1      ;DOUBLE SIDED MAX.
0350      32 04DA      ..EXIT: STA      TRK.MX      ;SET MAX TRACK.
0353      AF          XRA      A          ;SET ZERO FLAG.
0354      C9          RET          ;RETURN TO CALLER.

;*****

```

```

;*****
; SYSTEM TRACKS TRANSFER FUNCTION - ENTRY POINT *
;*****

;***** ( TRANSFER INITIALIZE )*****

0355    21 03A9    TRNSFR: LXI    H,ST.LST    ;ADDR OF TRANSFER LIST.
0358    22 04D2                SHLD    TF.PTR    ;SET TRANSFER POINTER.

;***** ( SET BIOS TRACK NUMBER )*****

035B    CD 0391    ..REPT: CALL    ..PLST    ;POP BYTE FROM LIST.
035E    FEFF                CPI    EQL    ;TEST FOR END OF LIST.
0360    C8                RZ    ;EXIT TRANSFER.
0361    CD 02BF                CALL    BS.TRK    ;BIOS SET TRACK.

;***** ( SET BIOS SECTOR NUMBER )*****

0364    CD 0391                CALL    ..PLST    ;POP BYTE FROM LIST.
0367    CD 02C2                CALL    BS.SEC    ;BIOS SET SECTOR.

;***** ( SET BIOS TRANSFER ADDRESS )*****

036A    CD 0391                CALL    ..PLST    ;POP BYTE FROM LIST.
036D    CD 039B                CALL    ..ADDR    ;CALCULATE ADDRESS.
0370    CD 02C5                CALL    BS.DMA    ;BIOS TRANSFER ADDR.

;***** ( SECTOR TRANSFER OPERATION )*****

0373    21 0386                LXI    H,..RETN    ;LOAD RETURN ADDRESS.
0376    E5                PUSH    H    ;PUSH ONTO STACK.
0377    3A 04D4                LDA    TF.DIR    ;LOAD TRNSFR DIRECTION.
037A    FE57                CPI    'W'    ;SEE IF WRITE FUNCTION.
037C    CA 02CB                JZ    BS.WRS    ;BIOS WRITE SECTOR.
037F    FE52                CPI    'R'    ;SEE IF READ FUNCTION.
0381    CA 02C8                JZ    BS.RDS    ;BIOS READ SECTOR.
0384    3EFF                MVI    A,ONES    ;ERROR CODE NOT R/W.
0386    B7    ..RETN: ORA    A    ;SET CONDITION CODES.
0387    CA 035B                JZ    ..REPT    ;DO SOME MORE.

;***** ( ENCOUNTERED DIFFICULTY )*****

038A    11 092B                LXI    D,MSG.NC    ;MESSAGE ADDRESS.
038D    CD 0297                CALL    MSG.OT    ;SEND MESSAGE.
0390    C9                RET    ;GO HOME.

;*****

```

FORMAT - JADE DOUBLE D

SYSTEM TRACKS TRANSFER SUBROUTINE

```

;*****
; POP LIST SUBROUTINE
;*****

```

```

0391      2A 04D2      ..PLST: LHL      TF.PTR      ;LOAD LIST POINTER.
0394      4E          MOV      C,M          ;GET ITEM NUMBER.
0395      23          INX      H          ;INCREMENT POINTER.
0396      22 04D2      SHLD     TF.PTR      ;STORE LIST POINTER.
0399      79          MOV      A,C          ;MOVE C TO ACUM.
039A      C9          RET                ;RETURN TO CALLER.

```

```

;*****
; GET MEMORY ADDRESS SUBROUTINE
;*****

```

```

039B      A7          ..ADDR: ANA      A          ;CLEAR CARRY BIT.
039C      1F          RAR                ;DIVIDE BY 2.
039D      47          MOV      B,A          ;HI ORDER TO B REG.
039E      3E00        MVI      A,0          ;CLEAR ACUMULATOR.
03A0      1F          RAR                ;CARRY BIT TO MSB.
03A1      4F          MOV      C,A          ;LO ORDER TO C REG.
03A2      2A 04D0      LHL      TF.INX      ;LOAD TRANSFER INDEX.
03A5      09          DAD      B          ;ADD IN OFFSET.
03A6      44          MOV      B,H          ;HALF MOV BC,HL.
03A7      4D          MOV      C,L          ;THE OTHER (HA!)
03A8      C9          RET                ;RETURN TO CALLER.

```

```

;*****

```

```

; *****
; THE FOLLOWING IS A LIST OF SYSTEM TRACK SECTORS *
; USED BY THE TRANSFR SUBROUTINE.  THERE ARE THREE *
; ENTRIES PER SECTOR.  1ST IS TRACK NUMBER.  2ND IS *
; SECTOR NUMBER.  3RD IS MEMORY LOAD OFFSET.      *
; *****
; SECTORS 2 THRU 26 ARE TRANSFERED ON TRACK 0.  SECTOR *
; 1 IS NOT TRANSFERED,  THIS IS THE  IDENTITY SECTOR. *
; TRACK 0 SECTOR ARE LOCATED IN SEQUENTIAL ORDER,  SO *
; THIS LIST IS STAGGERED.  SECTORS 1 THRU 48 ARE *
; TRANSFERED ON TRACK 1.                          *
; *****

0000      TK0      ==      0      ;DEFINE TRACK 0.
0001      TK1      ==      1      ;DEFINE TRACK 1.
00FF      EOL      ==      OFFH    ;DEFINE END OF LIST.

; *****

03A9      000404000808 ST.LST: .BYTE  TK0,04,04,TK0,08,08,TK0,12,12,TK0,16,16
03B5      001414001818      .BYTE  TK0,20,20,TK0,24,24,TK0,02,02,TK0,06,06
03C1      000A0A000E0E      .BYTE  TK0,10,10,TK0,14,14,TK0,18,18,TK0,22,22
03CD      001A1A000505      .BYTE  TK0,26,26,TK0,05,05,TK0,09,09,TK0,13,13
03D9      001111001515      .BYTE  TK0,17,17,TK0,21,21,TK0,25,25,TK0,03,03
03E5      000707000B0B      .BYTE  TK0,07,07,TK0,11,11,TK0,15,15,TK0,19,19
03F1      001717      .BYTE  TK0,23,23

03F4      01011B01021C      .BYTE  TK1,01,27,TK1,02,28,TK1,03,29,TK1,04,30
0400      01051F010620      .BYTE  TK1,05,31,TK1,06,32,TK1,07,33,TK1,08,34
040C      010923010A24      .BYTE  TK1,09,35,TK1,10,36,TK1,11,37,TK1,12,38
0418      010D27010E28      .BYTE  TK1,13,39,TK1,14,40,TK1,15,41,TK1,16,42
0424      01112B01122C      .BYTE  TK1,17,43,TK1,18,44,TK1,19,45,TK1,20,46
0430      01152F011630      .BYTE  TK1,21,47,TK1,22,48,TK1,23,49,TK1,24,50
043C      011933011A34      .BYTE  TK1,25,51,TK1,26,52,TK1,27,53,TK1,28,54
0448      011D37011E38      .BYTE  TK1,29,55,TK1,30,56,TK1,31,57,TK1,32,58
0454      01213B01223C      .BYTE  TK1,33,59,TK1,34,60,TK1,35,61,TK1,36,62
0460      01253F012640      .BYTE  TK1,37,63,TK1,38,64,TK1,39,65,TK1,40,66
046C      012943012A44      .BYTE  TK1,41,67,TK1,42,68,TK1,43,69,TK1,44,70
0478      012D47012E48      .BYTE  TK1,45,71,TK1,46,72,TK1,47,73,TK1,48,74

0484      FF      .BYTE  EOL      ;END OF LIST.

; *****

```

```
*****
; SELECT DRIVE THRU BIOS
*****
```

```
***** ( DISPLAY MESSAGE AND WAIT FOR RESPONSE ) *****
```

0485	22 04D5	SEL.DR: SHLD	MSG.SV	;SAVE MESSAGE ADDRESS.
0488	2A 04D5	..REPT: LHLD	MSG.SV	;LOAD MESSAGE ADDRESS.
048B	EB		XCHG	;PUT ADDRESS IN DE.
048C	CD 0297		CALL MSG.OT	;ISSUE MESSAGE.
048F	CD 029C		CALL CNS.IN	;CONSOLE INPUT.
0492	3A 0501		LDA RC.NBR	;LOAD NMBR OF CHARS.
0495	FE01		CPI 1	;SEE IF ONE CHARACTER.
0497	C2 0488		JNZ ..REPT	;IF NOT 1 CHAR TOO BAD.

```
***** ( SEE IF DRIVE LETTER GOOD ) *****
```

049A	3A 0502		LDA RC.TXT	;LOAD LETTER.
049D	D641		SUI 'A'	;GET NUMBER.
049F	DA 04B1		JC ..ILLG	;ILLEGAL, REPEAT.
04A2	FE04		CPI 04H	;DRIVE A THRU D?
04A4	DA 04BA		JC ..NMBR	;GOOD NUMBER.
04A7	D620		SUI 'A'-'A'	;OFFSET LOWER CASE.
04A9	DA 04B1		JC ..ILLG	;ILLEGAL, REPEAT.
04AC	FE04		CPI 04H	;LOWER A THRU D?
04AE	DA 04BA		JC ..NMBR	;LEGAL DRIVE.

```
***** ( EXIT TO RESELECT FUNCTION ) *****
```

04B1	11 07A2	..ILLG: LXI	D,MSG.SE	; 'SELECT ERROR'
04B4	CD 0297		CALL MSG.OT	;ISSUE MESSAGE.
04B7	C3 0488		JMP ..REPT	;REPEAT SELECTION.

```
***** ( VALID DRIVE NUMBER ) *****
```

04BA	32 04DE	..NMBR: STA	SV.NBR	;SAVE DRIVE NUMBER.
04BD	4F		MOV C,A	;DRIVE NMBR TO C.
04BE	1E01		MVI E,NO.LOG	;LOG ON VECTOR.
04C0	CD 02BC		CALL BS.DSK	;BIOS SELECT DISK.
04C3	7C		MOV A,H	;CHECK RETURN ADDR.
04C4	B5		ORA L	;SET FLAGS Z/NZ.
04C5	CA 04B1		JZ ..ILLG	;BIOS SAID NOGO IF 0.
04C8	3A 04DE		LDA SV.NBR	;GET NUMBER.
04CB	C9		RET	;RETURN CALLER.

```
*****
```

```

;*****
; SOFTWARE TRAPS - DDT EXIT
;*****

04CC    FF          RST.7:  RST      7          ;EXIT FORMAT.
04CD    C3 016A     JMP      SELECT        ;RETURN FOR SELECTION.

;*****
; WORKING VARIABLES
;*****

04D0    0F80        TF.INX: .WORD  0F80H    ;TRANSFER INDEX.
04D2    0000        TF.PTR: .WORD  0        ;LIST ADDRESS POINTER.
04D4    00          TF.DIR: .BYTE  0        ;TRANSFER DIRECTION.
04D5    0000        MSG.SV: .WORD  0        ;MESSAGE SAVE ADDRESS.
04D7    00          FT.STS: .BYTE  0        ;FORMAT STATUS SAVE.
04D8    00          TS.FLG: .BYTE  0        ;TWO SIDED DRIVE FLAG.
04D9    00          TRK.NO: .BYTE  0        ;TRACK NUMBER HOLD.
04DA    00          TRK.MX: .BYTE  0        ;LAST TRACK LIMIT.
04DB    00          SEC.NO: .BYTE  0        ;SECTOR NUMBER HOLD.
04DC    00          F.FLAG: .BYTE  0        ;FORMAT FLAG (DCM).
04DD    00          SYS.RF: .BYTE  0        ;SYSTEM TRACK READ FLAG.
04DE    00          SV.NBR: .BYTE  0        ;SEL.DV TEMP STORAGE.
04DF    00          FD.NBR: .BYTE  0        ;FORMAT DRIVE NUMBER.

04E0          STACK: .BLKW  16          ;PROGRAM STACK AREA.
0500          SP.TOP ==      .          ;TOP OF STACK.

;*****
; CONSOLE INPUT BUFFER AREA
;*****

0014          CB.SIZ ==      20          ;CONSOLE BUFFER SIZE SET.

0500    14          RC.BUF: .BYTE  CB.SIZ  ;DECLARE BUFFER SIZE.
0501    00          RC.NBR: .BYTE  0        ;INPUT STRING SIZE.
0502          RC.TXT: .BLKB  CB.SIZ  ;RESERVE CONSOLE BUFFER AREA.

;*****

```

```
*****
; JADE SINGLE DENSITY - IDENTITY SECTORS
*****
```

```
0516      4A6164652044 IDS.SS: .ASCII  "JADE DD S SIDED S DENSITY FORMAT "
```

0536		.LOC	IDS.SS+20H	;LOCATE CP/M 2.2 DPB.
0536	001A	.WORD	26	;SECTORS PER TRACK.
0538	03	.BYTE	3	;BLOCK SHIFT FACTOR.
0539	07	.BYTE	7	;BLOCK MASK.
053A	00	.BYTE	0	;EXM.
053B	00F2	.WORD	26*75/8-1	;DISK SIZE - 1.
053D	003F	.WORD	63	;DIRECTORY MAXIMUM.
053F	C0	.BYTE	11000000B	;ALLOC 0.
0540	00	.BYTE	0	;ALLOC 1.
0541	0010	.WORD	16	;CHECK SIZE.
0543	0002	.WORD	2	;TRACK OFFSET.

0546		.LOC	IDS.SS+30H	;LOCATE DCM BLOCK.
0546	00	.BYTE	0	;NOT USED.
0547	02	SD.FLG: .BYTE	00000010B	;DISKETTE FLAGS.

```
0596      .LOC      IDS.SS+SEC.SZ      ;EXTEND FULL SECTOR.
```

```
*****
```

```
0596      4A6164652044 IDS.DS: .ASCII  "JADE DD D SIDED S DENSITY FORMAT "
```

05B6		.LOC	IDS.DS+20H	;LOCATE CP/M 2.2 DPB.
05B6	001A	.WORD	26	;SECTORS PER TRACK.
05B8	04	.BYTE	4	;BLOCK SHIFT FACTOR.
05B9	0F	.BYTE	15	;BLOCK MASK.
05BA	01	.BYTE	1	;EXM.
05BB	00F6	.WORD	26*152/16-1	;DISK SIZE - 1.
05BD	003F	.WORD	63	;DIRECTORY MAXIMUM.
05BF	80	.BYTE	10000000B	;ALLOC 0.
05C0	00	.BYTE	0	;ALLOC 1.
05C1	0010	.WORD	16	;CHECK SIZE.
05C3	0002	.WORD	2	;TRACK OFFSET.

05C6		.LOC	IDS.DS+30H	;LOCATE DCM BLOCK.
05C6	00	.BYTE	0	;NOT USED.
05C7	0A	.BYTE	00001010B	;DISKETTE FLAGS.

```
0616      .LOC      IDS.DS+SEC.SZ      ;EXTEND FULL SECTOR.
```

```
*****
```



```
*****
; JADE DOUBLE DENSITY - IDENTITY SECTORS
*****
```

```
0616      4A6164652044 IDS.SD: .ASCII  "JADE DD S SIDED D DENSITY FORMAT "

0636                      .LOC      IDS.SD+20H      ;LOCATE CP/M 2.2 DPB.
0636      0032              .WORD      50              ;SECTORS PER TRACK.
0638      04                .BYTE      4              ;BLOCK SHIFT FACTOR.
0639      0F                .BYTE      00001111B      ;BLOCK MASK.
063A      01                .BYTE      1              ;EXM.
063B      00E9              .WORD      50*75/16-1      ;DISK SIZE - 1.
063D      003F              .WORD      63              ;DIRECTORY MAXIMUM.
063F      80                .BYTE      10000000B      ;ALLOC 0.
0640      00                .BYTE      0              ;ALLOC 1.
0641      0010              .WORD      16              ;CHECK SIZE.
0643      0002              .WORD      2              ;TRACK OFFSET.

0646                      .LOC      IDS.SD+30H      ;LOCATE DCM BLOCK.
0646      00                .BYTE      0              ;NOT USED.
0647      06                DD.FLG: .BYTE      00000110B ;DISKETTE FLAGS.

0696                      .LOC      IDS.SD+SEC.SZ    ;EXTEND TO FULL SIZE
```

```
*****
```

```
0696      4A6164652044 IDS.DD: .ASCII  "JADE DD D SIDED D DENSITY FORMAT "

06B6                      .LOC      IDS.DD+20H      ;LOCATE CP/M 2.2 DPB.
06B6      0032              .WORD      50              ;SECTORS PER TRACK.
06B8      05                .BYTE      5              ;BLOCK SHIFT FACTOR.
06B9      1F                .BYTE      31              ;BLOCK MASK.
06BA      03                .BYTE      3              ;EXM.
06BB      00EC              .WORD      50*152/32-1      ;DISK SIZE - 1.
06BD      007F              .WORD      127             ;DIRECTORY MAXIMUM.
06BF      80                .BYTE      10000000B      ;ALLOC 0.
06C0      00                .BYTE      0              ;ALLOC 1.
06C1      0020              .WORD      32              ;CHECK SIZE.
06C3      0002              .WORD      2              ;TRACK OFFSET.

06C6                      .LOC      IDS.DD+30H      ;LOCATE DCM BLOCK.
06C6      00                .BYTE      0              ;NOT USED.
06C7      0E                .BYTE      00001110B      ;DISKETTE FLAGS.

0716                      .LOC      IDS.DD+SEC.SZ    ;EXTEND TO FULL SIZE
```

```
*****
```

```
0716      ;*****
MSG.BG:      ;CONSOLE SIGN ON      *
;*****

0716      ODOA      .ASCII [CR][LF]
0718      ODOA2D2D2D2D .ASCII [CR][LF]'-----'
073A      ODOA464F524D .ASCII [CR][LF]'FORMAT UTILITY 2 - JADE DOUBLE D'
075C      ODOA2D2D2D2D .ASCII [CR][LF]'-----'
077E      ODOA24      .ASCII [CR][LF][EOM]

;*****
0781      MSG.FD:      ;FORMAT ON DRIVE SELECT      *
;*****

0781      ODOA53656C65 .ASCII [CR][LF]'SELECT DRIVE TO BE FORMATTED: '[EOM]

;*****
07A2      MSG.SE:      ;SELECTION ERROR      *
;*****

07A2      ODOA4E4F5420 .ASCII [CR][LF]'NOT A VALID SELECTION '[EOM]

;*****
07BB      MSG.FL:      ;DISPLAY FUNCTION SELECTIONS      *
;*****

07BB      ODOA0DOA      .ASCII [CR][LF][CR][LF]
07BF      ODOA2D2D2D2D2D .ASCII [CR][LF]'-----'
07E1      ODOA20202020 .ASCII [CR][LF]'          FUNCTIONS LIST      '
0803      ODOA2D2D2D2D2D .ASCII [CR][LF]'-----'
0825      ODOA      .ASCII [CR][LF]
0827      ODOA20312E20 .ASCII [CR][LF]' 1. FORMAT  DOUBLE DENSITY 8" '
0847      ODOA20322E20 .ASCII [CR][LF]' 2. FORMAT  SINGLE DENSITY 8" '
0867      ODOA20332E20 .ASCII [CR][LF]' 3. FORMAT  STANDARD 3740  8" '
0887      ODOA20342E20 .ASCII [CR][LF]' 4. READ    SYSTEM TRACKS IMAGE '
08AA      ODOA20352E20 .ASCII [CR][LF]' 5. WRITE   SYSTEM TRACKS IMAGE '
08CD      ODOA      .ASCII [CR][LF]
08CF      ODOA2D2D2D2D2D .ASCII [CR][LF]'-----'
08F1      ODOA24      .ASCII [CR][LF][EOM]

;*****
08F4      MSG.SF:      ;SELECT FUNCTION      *
;*****

08F4      ODOA454E5445 .ASCII [CR][LF]'ENTER FUNCTION NUMBER: '[EOM]

;*****
```

```
090E          ;*****
MSG.RS:      ;READ SYSTEM ON DRIVE      *
;*****

090E      0DOA      .ASCII [CR][LF]
0910      0DOA52454144 .ASCII [CR][LF]'READ SYSTEM FROM DRIVE: '[EOM]

;*****
092B      MSG.NC:    ;TRANSFER INCOMPLETE      *
;*****

092B      0DOA      .ASCII [CR][LF]
092D      0DOA5452414E .ASCII [CR][LF]'TRANSFER INCOMPLETE'
0942      0DOA24     .ASCII [CR][LF][EOM]

;*****
0945      MSG.FE:    ;FORMAT ERROR      *
;*****

0945      0DOA      .ASCII [CR][LF]
0947      0DOA464F524D .ASCII [CR][LF]'FORMAT TRACK ERROR'
095B      0DOA24     .ASCII [CR][LF][EOM]

;*****
095E      MSG.NR:    ;SYSTEM NOT LOADED      *
;*****

095E      0DOA      .ASCII [CR][LF]
0960      0DOA53595354 .ASCII [CR][LF]'SYSTEM TRACKS NOT LOADED'
097A      0DOA24     .ASCII [CR][LF][EOM]

;*****
```

```
*****
; FORMAT - TITLE BLOCK AND PAGE ALIGNMENT
;*****
```

```
      .DEFINE FORMAT [NAME] = [
NAME    ==      (,!OFFH)+1      ;SET NEXT PAGE BOUNDARY.
      .LOC      NAME            ;SET LOC TO NEXT PAGE.
OFFSET  =      FMT.EA-NAME      ;DETERMINE ADDR OFFSET.
      .Z80              ;NOW USE Z80 CODE.
      .ASCII 'FORMAT!']      ;INCLUDE HEADER!
```

```
*****
; DENSITY - DECLARE TYPE
;*****
```

```
      .DEFINE DENSITY [TYPE] = [
      .IFIDN [TYPE][SINGLE], [
      .ASCII 'S'
      .EXIT]
      .IFIDN [TYPE][DOUBLE], [
      .ASCII 'D'
      .EXIT]
      .ERROR 'INVALID DENSITY']
```

```
*****
; SECTORS - SPECIFY SEQUENCE AND NUMBER OF SECTORS
;*****
```

```
      .DEFINE SECTORS [LIST,NMBR] = [
      LXI      H,LIST+OFFSET      ;SECTOR SEQUENCE ADDR.
      MVI      E,NMBR            ;NUMBER OF SECTORS.
```

```
*****
; BLOCK - GENERATE A BLOCK OF CONSTANTS
;*****
```

```
      .DEFINE BLOCK [COUNT,BYTE,%REPT] = [
      NMBR = COUNT              ;SET EQUAL FOR NOW.
      MVI      B,NMBR          ;LOAD NMBR OF BYTES.
%REPT: IN      XP.DSH          ;WAIT FOR DATA REQ.
      MVI      A,BYTE          ;LOAD BYTE VALUE.
      XRA      C              ;INVERT (1791-01).
      OUT      WD.DTA          ;WRITE DATA PORT.
      DJNZ     %REPT]          ;REPEAT FOR COUNT.
```

```
*****
```

```
*****  
; REPEAT - REPEAT FORMAT SECTION FOR EACH SECTOR *  
*****
```

```
    .DEFINE REPEAT [LOCATION] = [  
        DCR     E                ;DEC NMBR SECTORS LEFT.  
        JNZ     LOCATION+OFFSET]
```

```
*****  
; ENDING - RECORD NMBR OF TRAILING BYTES WRITTEN *  
*****
```

```
    .DEFINE ENDING [BYTE,%REPT] = [  
        LXI     H,0                ;COUNT OF ZERO.  
%REPT:    IN     XP.DSH            ;WAIT FOR REQ.  
        MVI     A,BYTE            ;LOAD CONSTANT.  
        XRA     C                ;INVERT (1791-01).  
        OUT     WD.DTA            ;WRITE TO PORT.  
        INX     H                ;INCREMENT COUNT.  
        JMP     %REPT+OFFSET      ;CONTINUE.]
```

```
*****
```

```

;*****
; WRITE - WRITE SPECIFIC FORMAT BYTES
;*****

```

```

      .DEFINE WRITE [TYPE,VALU] = [

```

```

;***** ( ID ADDRESS MARK )*****

```

```

      .IFIDN [TYPE][ID.MARK], [
      IN      XP.DSH          ;WAIT FOR DATA REQ.
      MVI     A,OFEH          ;ID ADDR MARK.
      XRA     C               ;INVERT (1791-01).
      OUT     WD.DTA          ;WRITE DATA PORT.
      .EXIT]                 ;TERMINATE MACRO

```

```

;***** ( INDEX MARK )*****

```

```

      .IFIDN [TYPE][INDEX.MARK], [
      IN      XP.DSH          ;WAIT FOR DATA REQ.
      MVI     A,OFCH          ;INDEX MARK.
      XRA     C               ;INVERT (1791-01).
      OUT     WD.DTA          ;WRITE DATA PORT.
      .EXIT]                 ;TERMINATE MACRO

```

```

;***** ( DATA ADDRESS MARK )*****

```

```

      .IFIDN [TYPE][DATA.MARK], [
      IN      XP.DSH          ;WAIT FOR DATA REQ.
      MVI     A,OFBH          ;DATA ADDR MARK.
      XRA     C               ;INVERT (1791-01).
      OUT     WD.DTA          ;WRITE DATA PORT.
      .EXIT]                 ;TERMINATE MACRO

```

```

;***** ( CRC )*****

```

```

      .IFIDN [TYPE][CRC], [
      IN      XP.DSH          ;WAIT FOR DATA REQ.
      MVI     A,OF7H          ;GENERATE CRC.
      XRA     C               ;INVERT (1791-01).
      OUT     WD.DTA          ;WRITE DATA PORT.
      .EXIT]                 ;TERMINATE MACRO

```

```

;***** ( EXPLICIT BYTE VALUE )*****

```

```

      .IFIDN [TYPE][BYTE], [
      IN      XP.DSH          ;WAIT FOR DATA REQ.
      MVI     A,VALU          ;EXPLICIT VALUE.
      XRA     C               ;INVERT (1791-01).
      OUT     WD.DTA          ;WRITE DATA PORT.
      .EXIT]

```

```

;***** ( TRACK NUMBER )*****

```

```

      .IFIDN [TYPE][TRACK.NO], [
      IN      XP.DSH          ;WAIT FOR REQUEST.

```



```

      IN      WD.TRK      ;GET TRACK NMBR.
      OUT     WD.DTA      ;WRITE DATA PORT.
      .EXIT]

```

***** (SECTOR NUMBER)*****

```

      .IFIDN [TYPE][SECTOR.NO], [
      IN      XP.DSH      ;WAIT FOR REQUEST.
      MOV     A,M         ;SET SECTOR NMBR.
      XRA     C           ;INVERT (1791-01).
      OUT     WD.DTA      ;WRITE DATA PORT.
      INX     H           ;INC SEC-NMBR PNTR.
      .EXIT              ;TERMINATE MACRO]

```

***** (SIDE NUMBER)*****

```

      .IFIDN [TYPE][SIDE.NO], [
      IN      XP.DSH      ;WAIT FOR REQUEST.
      MVI     A,0         ;SET SIDE NUMBER.
      XRA     C           ;INVERT (1791-01).
      OUT     WD.DTA      ;WRITE DATA PORT.
      .EXIT              ;TERMINATE MACRO]

```

***** (SECTOR SIZE CODE)*****

```

      .IFIDN [TYPE][SECTOR.SIZE], [
      SEC.CD = OFFH      ;DECLARE BLANK.
      .IFIDN [VALU][128], [SEC.CD = 000H]
      .IFIDN [VALU][256], [SEC.CD = 001H]
      .IFIDN [VALU][512], [SEC.CD = 002H]
      .IFIDN [VALU][1024], [SEC.CD = 003H]
      .IFE     (SEC.CD-OFFH), [
      .ERROR   'INVALID SECTOR SIZE']
      IN      XP.DSH      ;WAIT FOR DATA REQ.
      MVI     A,SEC.CD    ;LOAD SIZE CODE.
      XRA     C           ;INVERT (1791-01).
      OUT     WD.DTA      ;WRITE DATA PORT.
      .EXIT              ;TERMINATE MACRO]

```

***** (ILLEGAL EXPANSION)*****

```

      .ERROR   'ILLEGAL EXPANSION']

```

0A00	464F524D4154	FORMAT	FT3740
0A07	53	DENSITY	SINGLE
0A08	21 17B5	SECTORS	SS3740,26
0A0D	0628	BG3740: BLOCK	40,ONES
0A18	0606	BLOCK	6,ZEROS
0A23	DB80	WRITE	INDEX.MARK
0A2A	061A	BLOCK	26,ONES
0A35	0606	RP3740: BLOCK	6,ZEROS
0A40	DB80	WRITE	ID.MARK
0A47	DB80	WRITE	TRACK.NO
0A4D	DB80	WRITE	SIDE.NO
0A54	DB80	WRITE	SECTOR.NO
0A5B	DB80	WRITE	SECTOR.SIZE,128
0A62	DB80	WRITE	CRC
0A69	060B	BLOCK	11,ONES
0A74	0606	BLOCK	6,ZEROS
0A7F	DB80	WRITE	DATA.MARK
0A86	0680	BLOCK	128,0E5H
0A91	DB80	WRITE	CRC
0A98	061B	BLOCK	27,ONES
0AA3	1D	REPEAT	RP3740
0AA7	21 0000	ENDING	ONES
0AB5	010203040506	SS3740: .BYTE	1, 2, 3, 4, 5, 6, 7, 8, 9,10
0ABF	0B0C0D0E0F10	.BYTE	11,12,13,14,15,16,17,18,19,20
0AC9	15161718191A	.BYTE	21,22,23,24,25,26

OB00	464F524D4154	FORMAT	FTJ50D
OB07	44	DENSITY	DOUBLE
OB08	21 17AE	SECTORS	SSJ50D,50
OB0D	0650	BGJ50D:	BLOCK 80,04EH
OB18	0608	RPJ50D:	BLOCK 8,ZEROS
OB23	0603		BLOCK 3,0F5H
OB2E	DB80		WRITE ID.MARK
OB35	DB80		WRITE TRACK.NO
OB3B	DB80		WRITE SIDE.NO
OB42	DB80		WRITE SECTOR.NO
OB49	DB80		WRITE SECTOR.SIZE,128
OB50	DB80		WRITE CRC
OB57	0616		BLOCK 22,04EH
OB62	060C		BLOCK 12,ZEROS
OB6D	0603		BLOCK 3,0F5H
OB78	DB80		WRITE DATA.MARK
OB7F	0680		BLOCK 128,0E5H
OB8A	DB80		WRITE CRC
OB91	0611		BLOCK 17,04EH
OB9C	1D		REPEAT RPJ50D
OBA0	21 0000	ENDING	ONES
OBAE	010B151F29	SSJ50D:	.BYTE 1,11,21,31,41
OB B3	020C16202A		.BYTE 2,12,22,32,42
OB B8	030D17212B		.BYTE 3,13,23,33,43
OB BD	040E18222C		.BYTE 4,14,24,34,44
OB C2	050F19232D		.BYTE 5,15,25,35,45
OB C7	06101A242E		.BYTE 6,16,26,36,46
OB CC	07111B252F		.BYTE 7,17,27,37,47
OB D1	08121C2630		.BYTE 8,18,28,38,48
OB D6	09131D2731		.BYTE 9,19,29,39,49
OB DB	0A141E2832		.BYTE 10,20,30,40,50
			.END

FORMAT - JADE DOUBLE D

+++++ SYMBOL TABLE +++++

BC.PTX 0009	BC.RCB 000A	BDOS 0005	BEGIN 0100
BG3740 0A0D	BGJ50D 0B0D	BS.DMA 02C5	BS.DSK 02BC
BS.FMT 02D4	BS.PTR 0001	BS.RDS 02C8	BS.SEC 02C2
BS.TRK 02BF	BS.VSZ 0033	BS.WRM 02A4	BS.WRS 02CB
B.MOVE 02D7	CB.SIZ 0014	CNS.IN 029C	CR 000D
DD.FLG 0647	EOL 00FF	EOM 0024	FD.NBR 04DF
FMT.DD 0321	FMT.EA 1700	FMT.SD 031B	FMT.ST 023B
FT3740 0A00	FTJ50D 0B00	FT.ERC 00FE	FT.STS 04D7
FT.TSM 0001	FUN.1 01AA	FUN.2 0208	FUN.3 01D2
FUN.4 01EB	FUN.5 025D	F.FLAG 04DC	IDS.DD 0696
IDS.DS 0596	IDS.SD 0616	IDS.SS 0516	ID.SEC 0001
INIT 0146	LF 000A	LIST 0164	MSG.BG 0716
MSG.FD 0781	MSG.FE 0945	MSG.FL 07BB	MSG.NC 092B
MSG.NR 095E	MSG.OT 0297	MSG.RS 090E	MSG.SE 07A2
MSG.SF 08F4	MSG.SV 04D5	NMBR 0011	NO.LOG 0001
OFFSET 0C00	ONES 00FF	RC.BUF 0500	RC.NBR 0501
RC.TXT 0502	REBOOT 0000	RP3740 0A35	RPJ50D 0B18
RST.7 04CC	SD.FLG 0547	SEC.CD 0000	SEC.NO 04DB
SEC.SZ 0080	SELECT 016A	SEL.DR 0485	SP.TOP 0500
SS3740 0AB5	SSJ50D 0BAE	STACK 04E0	ST.DMA 0324
ST.LST 03A9	SV.NBR 04DE	SYS.RF 04DD	TF.DIR 04D4
TF.INX 04D0	TF.PTR 04D2	TK0 0000	TK1 0001
TPA 0100	TRK.0 0000	TRK.1 0001	TRK.2 0002
TRK.ER 028E	TRK.MX 04DA	TRK.NO 04D9	TRK.NX 0279
TRANSFR 0355	TS.FLG 04D8	WDD.ID 02F2	WD.DTA 0007
WD.TRK 0005	WRT.ID 02FF	WSD.ID 02E2	XP.DSH 0080
ZEROS 0000			

DDT
DDT VERS 2.2
-IFORMAT.COM
-R

NEXT PC
0C00 0100
-DA00,AFF

```
0A00 46 4F 52 4D 41 54 21 53 21 B5 17 1E 1A 06 28 DB FORMAT!S!.....(.  
0A10 80 3E FF A9 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 .>.....>...  
0A20 07 10 F7 DB 80 3E FC A9 D3 07 06 1A DB 80 3E FF .....>.....>..  
0A30 A9 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 07 10 F7 .....>.....  
0A40 DB 80 3E FE A9 D3 07 DB 80 DB 05 D3 07 DB 80 3E ..>.....>..  
0A50 00 A9 D3 07 DB 80 7E A9 D3 07 23 DB 80 3E 00 A9 .....^...#...>..  
0A60 D3 07 DB 80 3E F7 A9 D3 07 06 0B DB 80 3E FF A9 .....>.....>..  
0A70 D3 07 10 F7 06 06 DB 80 3E 00 A9 D3 07 10 F7 DB .....>.....  
0A80 80 3E FB A9 D3 07 06 80 DB 80 3E E5 A9 D3 07 10 .>.....>.....  
0A90 F7 DB 80 3E F7 A9 D3 07 06 1B DB 80 3E FF A9 D3 ...>.....>...  
0AA0 07 10 F7 1D C2 35 17 21 00 00 DB 80 3E FF A9 D3 .....5.!.....>..  
0AB0 07 23 C3 AA 17 01 02 03 04 05 06 07 08 09 0A 0B .#.....  
0AC0 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 00 .....  
0AD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0AE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0AF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
-DB00,BFF
```

```
0B00 46 4F 52 4D 41 54 21 44 21 AE 17 1E 32 06 50 DB FORMAT!D!...2.P.  
0B10 80 3E 4E A9 D3 07 10 F7 06 08 DB 80 3E 00 A9 D3 .>N.....>...  
0B20 07 10 F7 06 03 DB 80 3E F5 A9 D3 07 10 F7 DB 80 .....>.....  
0B30 3E FE A9 D3 07 DB 80 DB 05 D3 07 DB 80 3E 00 A9 >.....>...  
0B40 D3 07 DB 80 7E A9 D3 07 23 DB 80 3E 00 A9 D3 07 .....^...#...>..  
0B50 DB 80 3E F7 A9 D3 07 06 16 DB 80 3E 4E A9 D3 07 ..>.....>N...  
0B60 10 F7 06 0C DB 80 3E 00 A9 D3 07 10 F7 06 03 DB .....>.....  
0B70 80 3E F5 A9 D3 07 10 F7 DB 80 3E FB A9 D3 07 06 .>.....>.....  
0B80 80 DB 80 3E E5 A9 D3 07 10 F7 DB 80 3E F7 A9 D3 ...>.....>...  
0B90 07 06 11 DB 80 3E 4E A9 D3 07 10 F7 1D C2 18 17 .....>N.....  
0BA0 21 00 00 DB 80 3E FF A9 D3 07 23 C3 A3 17 01 0B !.....>.....#.....  
0BB0 15 1F 29 02 0C 16 20 2A 03 0D 17 21 2B 04 0E 18 ..)....*....!+...  
0BC0 22 2C 05 0F 19 23 2D 06 10 1A 24 2E 07 11 1B 25 ",...#-...$....%  
0BD0 2F 08 12 1C 26 30 09 13 1D 27 31 0A 14 1E 28 32 /...&0...^1...(2  
0BE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
0BF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

